* Problem concentrated:

Normally when you interact with a web application, you use a web browser to connect to a central server over a network. All the code of this web application lives on this central server, and all the data lives in a central database. **Anytime you transact with your application, must communicate with this central server on the web.**

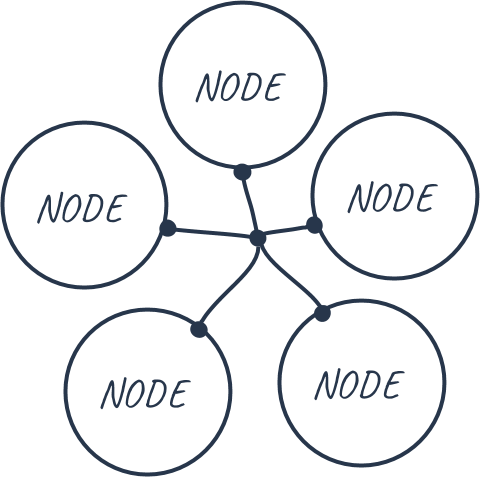
If we were to build our voting application on the web, we’d run into a few problems:

1. The data on the database could be changed: it could be counted more than once, or removed entirely.
2. The source code on the web server could also be changed at any time.

* Solution Discovered:

We don’t want to build our app on the web. **We want to build it on the blockchain where anyone connected to the network can participate in the election. We want to ensure that their votes are counted, and that they are only counted once.** So let’s take a look at how that works.

Instead of having a network, a central server, and a database, ***the blockchain is a network and a database all in one*.** A blockchain is a peer-to-peer network of computers, called nodes, that share all the data and the code in the network. So, if you’re a device connected to the blockchain, you are a node in the network, and you talk to all the other computer nodes in the network. **You now have a copy of all the data and the code on the blockchain.** There are no more central servers. Just a bunch of computers that talk to one another on the same network.



Instead of a centralized database, **all the transaction data that is shared across the nodes in the blockchain is contained in *bundles of records called blocks, which are chained together* to create the public ledger.** This public ledger represents all the data in the blockchain. All the data in the public ledger is secured by cryptographic hashing, and validated by a consensus algorithm. Nodes on the network participate to ensure that all copies of the data distributed across the network are the same. That’s one very important reason why we’re building our voting application on the blockchain, because we want to ensure that our vote was counted, and that it did not change.

What would it look like for a user of our application to vote on the blockchain? Well, for starters, the user needs an account with a wallet address with some Ether, Ethereum's cryptocurrency. Once they connect to the network, they cast their vote and pay a small transaction fee to write this transaction to the blockchain. **This transaction fee is called “gas”**. Whenever the vote is cast, some of the nodes on the network, called miners, compete to complete this transaction. The miner who completes this transaction is awarded the Ether that we paid to vote.

* ALGORITHM :

1. Create Smart Contract.
2. Deployed created contract on the local ethereum block chain.
3. Run migrate command to ensure deployed well.
4. Console to interact with smart contracts.
5. Write test cases for contracts providing account and etc.
6. Writing JavaScript library that help to interact with smart blockchain.
7. Writing render function to display contracts information.